NUWC-NPT Technical Document 10,260
1 September 1993

The Computer Virus: Infection, Removal, and Protection

W. P. Murray
Trident Command and Control Systems Department

Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

This document was prepared under Code 23 internal funding.

Reviewed and Approved: 1 September 1993

P. A. La Brecque
Head, Trident Command and Control Systems Department

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
1 September 1993

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
The Computer Virus: Infection, Removal, and Protection

5. FUNDING NUMBERS

6. AUTHOR(S)
W. P. Murray

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Undersea Warfare Center Division
1176 Howell Street
Newport, Rhode Island 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER
TD 10,260

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A new breed of computer programs -- commonly referred to as "viruses" -- has the capability not only of reserving malicious damage for a given event's occurrence but also of replicating itself. Typically, a virus program will spread by replicating a portion of itself onto another program. Later execution of that program will cause the virus to activate and spread to other programs. Once it is physically in a system, the virus can damage or destroy data, media, the system itself, and any attached peripherals.

This brief report describes the nature of the problems that virus programs pose; it also discusses common viruses, methods for virus detection and removal, and methods for protection against viral infections.

14. SUBJECT TERMS
Computers
Computer Viruses
Computer Programs

15. NUMBER OF PAGES
15

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

TABLE OF CONTENTS

THE COMPUTER VIRUS: INFECTION, REMOVAL, AND PROTECTION

INTRODUCTION

A computer virus is a malicious program usually hidden inside harmless looking "Trojan" carriers (named after the infamous Trojan horse of Greek mythology) and is usually stored on illegal, pirated copies of commercial program disks. The virus modifies other programs to include an executable (and possibly altered) copy of itself. Viruses usually pursue a sinister and long-range plan—the conversion of ordinary programs into infected Trojan carriers.

This document describes the problems that virus programs pose; it also discusses common viruses, methods for virus detection and removal, and methods for protection against virus infections. The discussion applies to Microsoft Disk Operating System (MS DOS) systems.

VIRUS PHASES

Most viruses have four phases; these are the dormancy phase, the propagation phase, the triggering phase, and the damaging phase. Virus creators might use the dormancy phase to instill a sense of trust in the user since the virus does not propagate or do damage during this phase. The propagation phase is all that is necessary for the program to be a virus program; a virus does not have to cause damage. The triggering phase is launched by some event or occurrence such as a particular date or a certain number of replications. Finally, the damaging phase does whatever harm the virus's author intended it to do.

Some viruses have pre-trigger coding that lets the virus sit benignly dormant until something transpires, such as a particular date or time frame, the presence of another program or file, or the disk capacity exceeding some predetermined limit. Generally speaking, most of the common viruses do not contain pre-triggers. Either a trigger or a pre-trigger activates the virus's replication mode, and the virus replicates a virtual copy of itself into other programs or into certain system areas on the disk. When loaded and executed, each infected program or system area is a clone of the virus itself and in turn will produce more clones, thereby spreading itself exponentially through systems and programs. Some viruses strike COMMAND.COM or hidden system files. Others modify the boot sector of floppy disks. Still others attach themselves to any .COM or .EXE file. In truth, any file that can be executed—whether it's a program, a device driver, an overlay, or even a batch file—can be the target of a virus.

VIRUS INFECTION PROCESS

While viruses can wreak their havoc in all computer environments, this document focuses on the MS DOS infection. Most viruses seen to date carry out their damage in an operating-system-specific manner. In the DOS environment, simple viruses infect the most basic programs of all—the .COM files. These programs are almost an exact image of the memory image of the program run after they are loaded. At load time, only after examination of the segment registers or the hard disk's file allocation tables, can such a program tell where it has been loaded in memory. Most COM programs start with a "jump instruction." If the jump instruction is modified so that it points to other code, the central processing unit (CPU) will automatically run that code. After the virus has done its dastardly deed, it must run the original program as if it were not infected. The virus need not be concerned with the meaning of the original data bytes it displaced to infect the errant program. It treats these bytes just like data, moving them back into their original location and allowing them to execute without regard as to what they are. Hence, even programs that do not start with jump instructions can be infected.

A virus program may be written in one of two ways: as position-independent code or position-dependent code. A position-independent virus is usually added to the end of a program. It has no need to manipulate more than the first three bytes of the original program to branch the program's run to itself. Position-independent programs are considered to be a little harder to write since, at run time, all data accesses must be localized and resolved. Position-dependent viruses must be set to execute at the start of the program and typically will copy the original instruction (s) at the beginning of the uninfected program to the physical end of the program's data image on disk. These viruses also contain a block-move routine that must be position independent so it can run away from any location or use a known and quantified area of memory outside the program's memory image, such as areas in the low-memory interrupt vector or video table. This is necessary to ensure that the original code displaced by the virus is restored to its proper location and will run correctly.

The essential component of a computer virus is a set of instructions that, when executed, spreads itself to other previously uninfected programs or files. A typical virus performs two functions. First, it copies itself into previously uninfected programs or files. Second, it executes whatever instructions the virus's author included in it. To manage their activities, viruses place coded bytes (called "v-markers" for virus markers) inside files during initial infections. When unmarked, uninfected files cannot be found, viruses assume that their hosts have been thoroughly poisoned with virus code. Depending on the motives of the virus's author, these instructions can do anything at all, including displaying taunting messages, erasing files, subtly altering stored data, or causing abnormal system behavior. In some cases, a virus may contain no harmful or disruptive instructions. Instead, it causes damage by replicating itself and using up scarce resources such as disk space, CPU time, or network connections. Eventually, however, most viruses detonate and damage or destroy disk-based data.

VIRUS-RELATED TERMINOLOGY

The information presented thus far establishes a small but basic foundation as to what a virus is and how it infects. Before continuing, it is important that virus "buzz words" or terminology be explained. Both computer users and the popular computer press often misuse the buzz words relating to the many faces of rogue software. Worms are confused with viruses; the three types of software bombs are treated as one all-encompassing classification, and so on. Table 1 presents a summary of some popular virus-related terminology.

Table 1. Some Virus-Related Terminology

Anti-virus	**A program designed to prevent or detect virus infections.**
Back Door	**Undocumented, secret program features or commands known (and perhaps available) only to the program's designer.**
Bomb	**A program that unconditionally executes destructive instructions without user authorization.**
Bug	**A hardware or software error that causes a system or program to function incorrectly.**
Drill Sergeant	**A bomb, logic bomb, or time bomb program that continuously executes other programs or operating system instructions.**
Logic Bomb	**A program that conditionally executes destructive instructions without user authorization, depending on the status of specific environmental variables. For example, a logic bomb could monitor payroll records, looking for the designer's social security number. The logic bomb may be programmed to detonate (erase payroll records or reformat the hard disk) if the designer's social security number fails to appear in the records for three consecutive weeks.**
Rabbit Replicator	**A bomb, logic bomb, or time bomb program that creates, continuously and without end, either on disk or in memory, independent, executable copies of itself.**
Rogue	**Any program specifically designed to destroy hardware or software without user authorization.**
Time Bomb	**A program that conditionally executes destructive instructions without user authorization, depending on the status of counters or time-related environmental variables.**
Trojan Horse	**A bomb, logic bomb, time bomb, virus, worm, or other rogue program that pretends to be an ordinary (nondestructive) program.**

Table 1. Some Virus-Related Terminology (Cont'd)

TSR	**Programs that load and stay RAM-resident (TSR is an MS DOS acronym for "terminate and stay resident").**
Virus	**A program that modifies other programs to include an executable (and possibly altered) copy of itself.**
Virus Detector	**A program designed to detect virus infections.**
Worm (Core)	**A program that consumes memory by traveling through it, much like a worm burrows through earth.**
Worm (Network)	**A program that travels through a network from computer to computer, sometimes creating executable copies of itself in the process.**

DETECTING VIRUS INFECTIONS

The two most important resources available for detecting virus infections are watchful users and watchful programs. The best detection approaches use both.

WATCHFUL USERS

As users are aware of the need for backups, they should also be aware of the possibility of viruses and what kind of things to watch for. Some of these things occur once a virus is already in place, others while the virus is still spreading. Users should be vigilant for things like the following:

- Unexpected changes in the time stamps or length of files, particularly executable files.
- Programs taking longer to start or running more slowly than usual.
- Programs attempting to write to write-protected media for no apparent reason.
- Unexplained decreases in the amount of available memory or increases in areas marked as "bad" on magnetic media.

•Executable files unexpectedly vanishing.

•Workstations unexpectedly rebooting at a relatively constant interval after being turned on or when previously correct programs are run.

•Unusual things appearing on the display like bouncing balls, taunting messages, or scrolling of odd parts of the screen.

•Unexpected changes to volume labels on disks and other media.

•An unusual load on local networks or other communication links, especially when multiple copies of the same data are being sent at once.

Not all odd behavior is necessarily indicative of virus presence. Software bugs, user errors, and hardware failures are common causes of these types of oddities. Users should be aware of oddities in general and avoid unfounded rumors of virus infection.

DETECTION PROGRAMS AND SOFTWARE PACKAGES

As previously stated, users are not the only line of defense in detecting viruses. It is comparatively simple to create programs that will detect the presence and spread of the simpler classes of viruses. One approach involves notifying the user of changes to the contents of executable objects such as program files, boot records, and so forth. A user can run a program once a day that will examine executable objects and compare their characteristics with those found the last time the program was run. The user can then be presented with a list of objects that have changed. If things have changed that should not have or if certain objects such as operating system files that seldom or never change are different, the user is being warned that a virus infection may have occurred. Another virus detection method is the use of detection software packages that range from freeware to shareware to commercial programs costing many hundreds of dollars. Many packages are available for different machines, and they vary in their abilities to detect virus infections.

Well-written computer viruses are difficult to detect using file management and anti-Trojan techniques. Viruses are simple yet ingenious programs that pollute systems by inserting copies of themselves into, appending viral clones onto, or creating shells around ordinary executable files. Detection products can only warn users after a virus infection has occurred, since they look for virus effects. These products check executable files, boot sectors, and so on, comparing them with a previously recorded signature. This signature is created by feeding every byte of a file into an algorithm (usually secret) that yields a large number. The probability that anyone could change the program without changing the signature is infinitesimal. Some products check the signatures of all files, while others check each program as it is run. Most of these detection products are capable of detecting the most common viruses (see table 2).

*Table 2. Twelve Common Viruses**

Pakistani Brain
Jerusalem
Alameda
Cascade (1701/1704)
Ping Pong
Stoned
Lehigh
Den Zuk
Datacrime (Columbus Day/Friday 13th)
Fu Manchu
Vienna (DOS 62)
April First

***The strains or subvariations of these viruses account for more than 90 percent of all reported PC infections.**

New viruses are being created every month, and most detection software packages are updated frequently to keep up with the growing threat. Some good detection software is distributed as shareware and may be licensed for a very moderate fee. Once a user is registered for such detection programs, upgrade notification is received and upgrades are available for a fraction of the initial cost. In addition to detecting the common viruses identified in table 2, some detection programs are capable of detecting viruses that have not yet caused widespread problems in the United States but are known to exist in Europe.

The frequency with which virus detectors should be used depends on the rate at which executables are transmitted and the value of the information assets on the system. Viruses that propagate by way of floppy disks do so relatively slowly. It may take days, weeks, or even months for a virus to propagate across a large organization. Running detection programs once a week may be adequate in these circumstances.

On the other hand, for systems connected to networks, especially large-scale networks, viruses can spread very rapidly. In some cases, replicating programs have spread nationwide in a matter of day or even hours. So, in these circumstances, it may be important to run virus detection programs hourly or daily. Since there is no perfect defense against computer viruses that still allows programming and sharing of executable information, defenses will have to be improved to deal with new classes of viruses. With the possible exception of isolation, the methods previously mentioned for detection and prevention are only somewhat reliable. Viruses can still infect some systems despite the implementation of preventive measures. Early detection of virus infection is still the best way to stop the spread. Because viruses can spread so quickly, it is very important to detect them as early as possible after the first infection. The surest way of doing this is to have every vulnerable user watch for the signs of infection and, when feasible, run the best available virus detection software as often as possible.

REMOVING VIRUS INFECTIONS

When a virus is detected, every moment spent deciding what to do next gives the virus more time to spread. Measures should be taken to stop the spread as soon as possible, initiate recovery, and get back to normal operations. Procedures should be in place for some basic functions such as the following:

- Identifying infected systems.
- Determining how and from where the infection reached each system known to be infected.
- Isolating infected systems until they can be rendered free of the virus.
- Informing vulnerable users about the virus and how to avoid spreading it.
- Removing the virus and identifying it.
- Developing or obtaining specific software or procedures to combat the virus (these may remove the virus from infected systems and files, determine whether or not backups are infected, etc.).
- Recording the characteristics of the virus and the effectiveness of the steps taken against it in case of later re-infection.

The main objective of the actions listed above is to provide the user with a normal, uninfected computing environment. Once the virus has been studied, the user can write or obtain programs to help remove it. Once the virus is understood in terms of the types of objects it spreads itself to, three categories of objects may be considered for restoration:

1. Objects of the type that the virus infects should be restored only from backups that have been thoroughly and individually checked for infection. When possible, restoration should be accomplished from trusted sources such as original, unwriteable copies provided by the manufacturer or by recompiling source code.

2. Objects of the type that the virus does not infect but that have been destroyed or modified by the virus's action may generally be restored from backups. Again, the integrity of backups should be verified because if the virus has been in the system for some time it may have damaged objects that were later backed up.

3. Objects of the type that the virus neither infects nor damages may or may not need to be restored. Efforts should be made to ensure that such objects have not been infected before deciding whether or not to restore them.

Virus removal programs should be used only when there is no other practical way to obtain an uninfected version of an object. Removing a virus from an executable object can be a complex and difficult process, and even well-written programs may fail to restore an object correctly.

Anti-virus programs or vaccines are products that actually inject themselves into executable programs. They are like a benign virus. When the program starts, the injected code performs a signature check and warns the user if any changes have been made. Unfortunately, by the time the user gets the warning, the virus has already infected the system and the virus code has already been activated. Some anti-virus programs or vaccines try to identify known viruses by looking for specific byte patterns. This approach is not very practical because a small change to the virus code can defeat the vaccine. Any useful anti-virus or vaccine program should offer damage control, both preventive and restorative. Preventive methods include stopping attempts at direct disk access, warning the user about programs that unexpectedly insert RAM-resident code into memory, and even write-protecting the hard disk while unfamiliar software is being used or tested. If a disk is formatted by a virus, the user may be able to reverse the damage if his anti-virus program maintains a copy of the file allocation table. Unless the virus has performed a low-level format, the data are still there.

Some anti-virus programs offer helpful features such as saving a copy of the CMOS memory. A user can probably employ this to restore his setup after the battery goes dead. It may also warn the user when he accidentally starts to copy one executable file over another with the same name. However, there are some drawbacks as well. The user's anti-virus program may not allow him to run applications that store default values by changing the program file and low-level disk editor programs. Also, vaccinated programs take longer to run because appended anti-virus code and data increase overhead; large amounts of disk space can be consumed as the appended anti-virus code and data enlarge program files; and data and overlay files cannot be protected. Many vaccines cannot protect packed .EXE files because they have been compressed during the programming LINK process to conserve disk space and they expand in memory when run; the source code modifications associated with these changes are often misinterpreted as virus alterations. The virus-like behavior of vaccines may cause conflict with other virus defense systems, and some viruses can detect the modification of target files and they simply delete the anti-virus code and data. Although anti-viruses and vaccines operate similarly to viruses, they do not reproduce without authorization and they do not purposely damage files. Most users are uncomfortable with injecting a virus (benign or otherwise) into their executable files especially when safer and less drastic alternatives exist.

While some anti-virus or vaccine vendors' products can remove viruses or repair virus-damaged files, the most reliable method is to restore files from certified backups or master

copies. Some vendors boldly claim their program's ability to restore virus-deleted data from reformatted and/or repartitioned hard disks. In truth, these products merely restore backup copies of critical disk information such as boot sectors, file allocation tables, disk directories, and partition data. Such products are known as "antidotes," and like their cousin, the anti-virus or vaccine program, they are not without their drawbacks. Most antidotes cannot reliably recover data if new data have been written to disk after the disk has been reformatted. If backup data are not current, the information the backup replaces will be out of date—an inaccuracy that may lead to further data loss. Antidotes or format recovery programs cannot reconstruct data erased by low-level or destructive high-level reformatting (low-level formatting is employed by most hard disk manufacturers and destructive high-level formatting is a feature of AT&T's and Compaq Computer's DOS and perhaps other DOS versions). It is safer and more reliable to restore files from certified backup copies or through the use of proven data recovery programs.

PROTECTING AGAINST VIRUS INFECTIONS

The best ways for users to protect against viruses are to apply common sense and to observe the following operating guidelines:

1. Always make frequent backups; multiple sets that can be rotated each day of the week are advisable. Multiple backups can help not only in recovery from malicious virus programs but also in recovery from hard disk crashes and accidental reformatting.

2. If it is determined conclusively that a virus infection has occurred and that data have been destroyed, reboot the system from a write-protected DOS disk, format the hard disk, and selectively restore only the data from backups. To reload programs, use original disks and reinstall them.

3. Always put write protect tabs on floppy disks that don't need to have data written to them. If a "write protect error" occurs, something improper is going on.

4. Use only established bulletin board systems. Although bulletin boards haven't been implicated in a major virus outbreak to date, they are a potential source of viruses.

5. Make all .COM and .EXE files read-only files.

6. Move COMMAND.COM out of the root directory. Hide it or build a shell around it.

7. Don't loan out program disks. If a loan must be made, make a DISKCOPY and format the disk when you get it back.

8. For floppy drive systems, use only one write-protected booting disk.

9. Don't let others use your computer system. If this is not possible, at least don't let others use their own program disks.

10. Don't use illegal copies of programs, especially if they have been "hacked" to remove copy protection.

SUMMARY

Virus programs pose a real threat in today's computing world. The potential for virus infection is high and new viruses are being developed each month. Watchful users and watchful programs are the first line of defense against these viruses.

Early detection is the key to reducing the spread of a virus. Wherever possible, conduct frequent, periodic virus checks with the best detection software available, especially among users who do large amounts of information exchange and those who generally run new software before others do. If a virus is detected, take immediate and positive measures to stop it from spreading. Take every step available to isolate and eradicate all evidence of the virus. Ensure that no infected objects remain. Restore files using only certified backups and programs; this is the key to preventing reinfection. Make frequent backups, as they can help immensely. Don't use illegal or questionable software without certifying that it is virus-free—the majority of infections to date have occurred from "hacked" illegal or pirated software. Watch out for reinfection from a particular virus.

Educate users about possible virus warning signs, and promote user awareness of methods to respond to virus infections. Much information pertaining to viruses, their effects, isolation, remedial action, etc., is available (see the Bibliography for some suggested sources). Last but not least, have a plan in place to deal with virus infections before they occur.

BIBLIOGRAPHY

Cohen, Fred, "Computer Viruses: Theory and Experiment," *Computers & Security*, vol. 6, 1987. pp. 22-35.

Cohen, Fred, "On the Implications of Computer Viruses and Methods of Defense," *Computers & Security*, vol. 7, 1988, pp. 167-184.

Greenberg, Ross M., "FLU SHOT+ Version 1.0, A Form of Protection from Viral and Trojan Programs," *Software Concepts Design,* 1988.

Greenberg, Ross M., "Know Thy Viral Enemy," *BYTE*, vol. 14, no. 6, June 1989, pp. 275-280.

Levin, Richard, "CHECKUP: Virus Detection System Documentation for Version 3.6," July, 1989.

Murray, W. H., "Security Considerations for Personal Computers," *IBM System Journal*, vol. 23, no. 3, 1984, pp. 297-304.

Murray, W. H., "Security Risk Assessment in Electronic Data Processing Systems," IBM Publication No. G320-9256-0, International Business Machines Corp., 1984.

Rubenking, Neil J., "Anti-Virus Programs: Is the Cure Worse Than the Disease?," *PC Magazine*, vol. 8, no. 8, April 1989.

White, Steve R., David M. Chess, and Chengi Jimmy Kuo, "Coping with Computer Viruses and Related Problems," IBM Research Report RC 14405, International Business Machines Corp., 30 January 1989.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Naval Sea Systems Command (PMS-396E23, SEA-09T4 (D. Corradino))	2
Defense Technical Information Center	2